Hızlı Hesaplama ve Mantık Kod Sistemi Veta Numaraları*

Süleymanov Ş. Elvin
16.11.2022

* Veta sayı sistemi - benzer sayılarla ancak farklı mantıksal aralıklarla çalışmak için harika fırsatlar sağlayan özel bir sayı sistemidir.

Anahtar kelimeler: mantık sistemi, mantıksal süreçler, aralık sayıları, veta sayısı, hızlı hesaplama sistemi, veta sistemi.

I. Gösterimler

1. V - veta numarası.
2. Nv - veta sayıları kümesi.
3. iv - veta numarası için belirtilen aralık.
4. V[0,0] ; (0,0) - Yanlış'ı Yanlış olarak hedefleyen iv aralığındaki değeri depolayan sayı.
5. V[0,1] ; (0,1) - True'yu hedefleyen iv aralığındaki değeri depolayan sayı.
6. V[1,1] ; (1,1) - True'yu True olarak hedefleyen iv aralığındaki değeri saklayan sayı.
7. V[1,0] ; (1,0) - Sıfır'ı hedefleyen iv aralığındaki değeri depolayan sayı.
8. f(x)vd - varsayılan sonucu döndüren veta işlevi.
9. f(a,b)v - iki giriş argümanı ile işlev.
10. VA, V-1 - ters veta sayısı.
11. Tv - veta numarası için belirtilen süre.
12. Xsi - veta sayısı yaklaşık değerleri kümesi.
13. η - hesaplama işlemi operatörü (+,-,*,:).

II. Giriiş

Şu anda, hesaplamalı mantık süreçleri gibi birçok süreci matematikte temsil etmek zordur ve ayrıca mantık istatistiklerine dayalı olarak gelecekteki büyümeyi tahmin etmek de zordur. Şimdi programcıların ve bilim adamlarının çoğu matematikte olduğu kadar programlama dillerinde de daha iyi bir AI mantıksal süreçleri kavramı geliştirmeye çalışıyor. Şu anda sahip olduğumuz her şey "eğer" veya "else" gibi basit terimler veya "0" veya "1" gibi bir sonuç elde etmenin basit bir yoluna sahibiz, ayrıca bulanık sayılarla bulanık mantığa sahibiz ve bu da en iyi fırsatı veriyor. çalışır, çünkü [0, 1] aralığındaki sayıları kullanırız. Ancak programlama ve matematik kullanımı için çok yavaş. Bu yüzden Veta numarasını oluşturdum. Bu, bir karakterli, ancak içinde farklı bir aralık bulunan bir sayıdır. Bu sayı, en iyi mantıksal hesaplamaya ve gelecekteki süreçlerin en iyi şekilde ele alınmasına olanak tanır. Veta sayısı en iyi, diğer arabaların arasında bir araba ile çalıştığımız ve bu arabanın kendi aralığına ve konfigürasyonuna sahip olduğu yoğun trafik olarak tanımlanabilir.

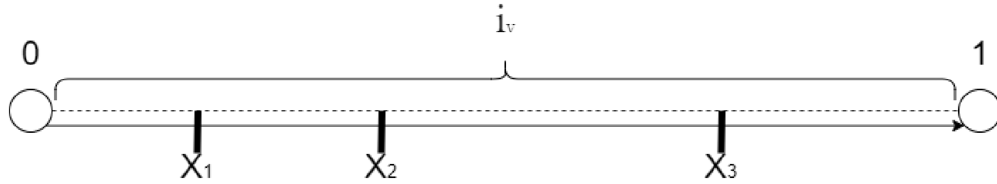
III. Veta Numarası

Veta numarasına bakalım. Veta Numarası - kendi tanımlı aralığından bir değişken değeri saklayan bir sayıdır.

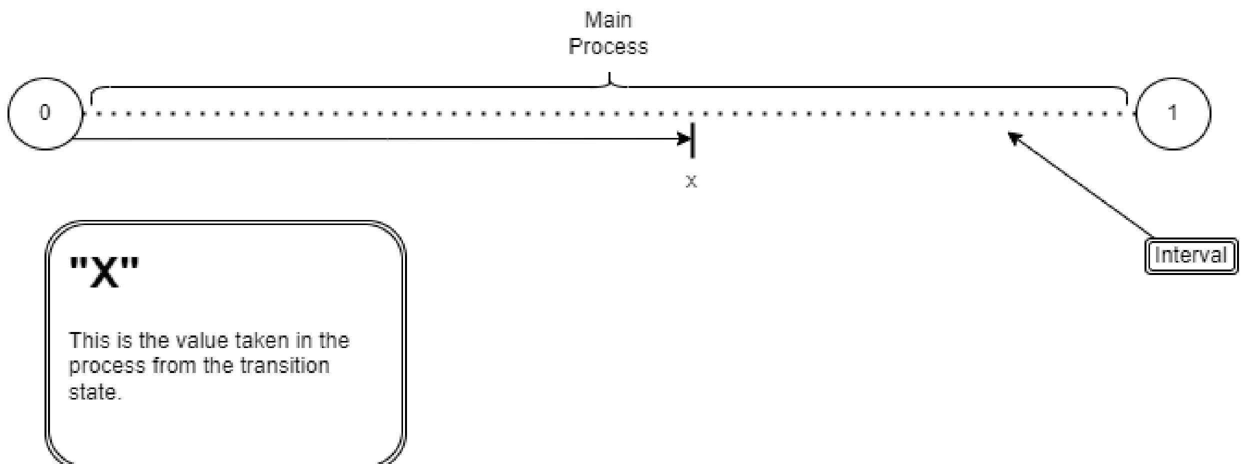
N ∈ {0,1} ∈
R N1 ∈ {1} ∈
R+ N0 ∈ {0}
∈ R-
Nv ∈ {(0,0), (1,1), (0,1), (1,0)}

1	Doğru (Değer: 1)
0	Yanlış (Değer: 0)
(0,1)	0 … 1'den alınan değer
(1,0)	1 … 0'dan alınan değer
(0,0)	0 … 0'dan alınan değer
(1,1)	1 … 1'den alınan değer

Bir şeyin bir şey olacağı tüm süreçler bir veta numarası ile yazılır. Örneğin, başlangıcı negatif "0" ve sonu pozitif veya "1" gibi olan bir sürecimiz var, onu tanımlamamız gerekiyor. Bir veta numarası (0,1) kullanmalıyız. (0,1) bu şekilde gösterilebilir:



Burada iv - gösterim aralığının uzunluğu, X1, X2, X3 - alınan veta sayılarıdır. Olası tüm veta sayılarının ana görünümü benzerdir, ancak iç karakterler aralıklar gibi farklı olabilir.
iv ∈ R+ ve burada iv ≥ 0. Herhangi bir zaman aralığı veta sayıları yaklaşık değere eşittir. Veta sayıları, varsayılan olarak iv'e eşit olan kişisel bir döneme sahiptir. Periyodun veta sayısı aralığıyla çarpımı, veta sayısı ilerlemelerinin en iyi görünümünü verir.



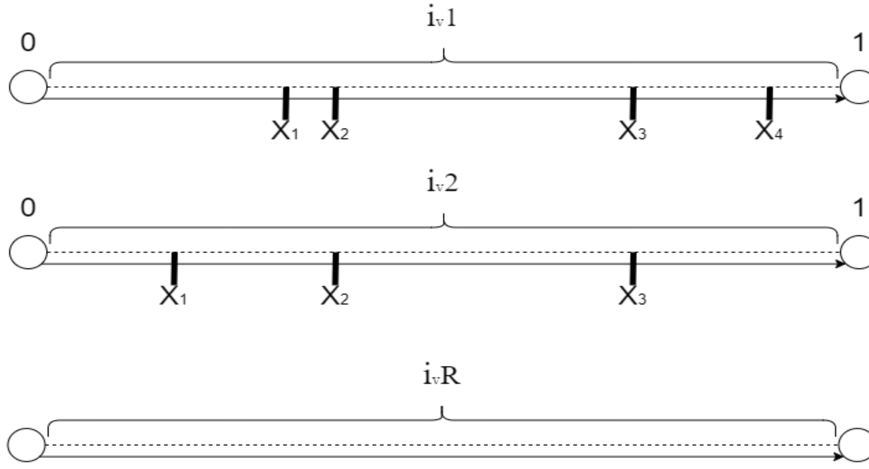
IV. Veta Numaraları ile Hesaplamalar

Hesaplama süreçleri bu sistemin en iyi parçasıdır. Hesaplamalar veta sayıları ile desteklenecek. Ayrıca iki veya daha fazla bağımsız değişken işlevi ve ayrıca Varsayılan ve Kaotik işlevler gibi yeni bir kavram tanıtacağız.

Hesaplama İşlemleri:

1. Ekleme
2. Çıkarma
3. Çarpma
4. Bölüm

Tüm hesaplama işlemleri mantıksal nedenlerle işlemek için gereklidir. Örneğin: doğru + doğru = doğru, 1 + 1 = 1.

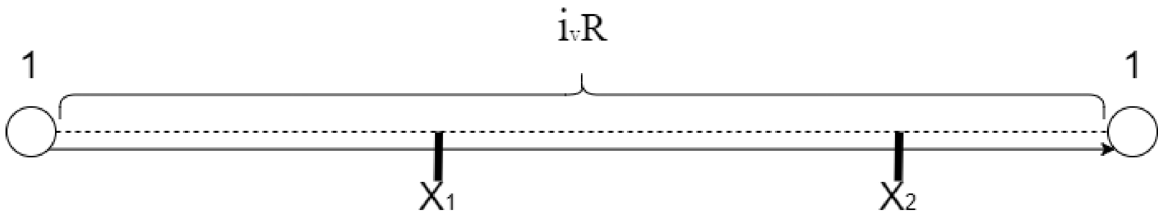


Hesaplama:

1. (iv1 + iv2) : 2 = ivR
2. Birinci ve İkinci grafikler (0,1) rakamlarını göstermektedir.
3. (0,1) + (0,1) = (1,1)
4. Xs1 ∈ {X1, X2, X3, X4}
5. Xs2 ∈ {X1, X2, X3}
6. XsR ∈ {X1 = (X1 + X2 + X3 + X4) : 4, X2 = (X1 + X2 + X3) : 3}

Sonuç olurdu:

Nv = (1,1) {(ivR), (X1, X2)}



V. Ek Veta Numaraları

+	0	1	(0,0)	(1,1)	(1,0)	(0,1)
0	0	(0,1)	0	(0,1)	0	0
1	(1,0)	1	(1,0)	1	1	1
(0,0)	0	(0,1)	0	(0,1)	(0,0)	(0,0)
(1,1)	(1,0)	1	(1,0)	1	(1,1)	(1,1)
(1,0)	0	(1,1)	0	(1,1)	(0,0)	(0,1)
(0,1)	(0,0)	1	(0,0)	(1,1)	(1,0)	(1,1)

$$a \eta b \eta c \neq c \eta b \eta a$$
$$1 + 0 + 0 \neq 0 + 0 + 1$$
$$1 + 0 + 0 = (1,0) = 0$$
$$0 + 0 + 1 = (0,1) = 1$$
$$0 \neq 1$$

(Subtraction, Multiplication, Division)
VI. Çıkarma Veta Sayıları

-	0	1	(0,0)	(1,1)	(1,0)	(0,1)
0	NULL	NULL	NULL	NULL	NULL	NULL
1	1	0	1	0	(0,1)	(1,0)
(0,0)	NULL	NULL	NULL	NULL	NULL	NULL
(1,1)	(1,1)	0	(1,1)	0	(0,1)	(1,0)
(1,0)	NULL	NULL	NULL	NULL	NULL	NULL
(0,1)	(0,1)	0	(0,1)	(1,0)	(1,0)	0

VII. Çarpma Veta Numaraları

*	0	1	(0,0)	(1,1)	(1,0)	(0,1)
0	0	(0,1)	0	(0,1)	(0,0)	(0,1)
1	(1,0)	1	(1,0)	1	(1,1)	(1,1)
(0,0)	0	(0,1)	0	(0,1)	(0,0)	(0,1)
(1,1)	(1,0)	1	(1,0)	1	(1,0)	(1,1)
(1,0)	0	(0,1)	0	(0,1)	0	(0,1)
(0,1)	(1,0)	1	(1,0)	1	(1,0)	1

VIII. Bölüm Veta Numaraları

:	0	1	(0,0)	(1,1)	(1,0)	(0,1)
0	0	0	0	0	0	0
1	NULL	1	NULL	1	NULL	NULL
(0,0)	0	0	0	0	0	0
(1,1)	NULL	1	NULL	1	NULL	NULL
(1,0)	0	0	0	0	0	0
(0,1)	NULL	1	NULL	(0,1)	NULL	1

IX. Uyum Operasyonu

Olası tüm basit hesaplama işlemleri yukarıda verilmiştir. Hesaplamanın ana kuralı, son basamağa dikkat etmemizdir.

a, b, c, d ∈ Nv (veta numaralar)

Örnek:

$$a \eta b \eta c \neq c \eta b \eta a$$

$$1 + 0 + 0 \neq 0 + 0 + 1$$

$$1 + 0 + 0 = (1,0) = 0$$

$$0 + 0 + 1 = (0,1) = 1$$

$$0 \neq 1$$

Ancak her işlemde son sayı benzer olduğunda:

$$a \eta b \eta c \eta d = c \eta b \eta a \eta d$$

$$1 + 0 + 0 + 1 = 0 + 0 + 1 + 1$$

$$1 + 0 + 0 + 1 = (1,1) = 1$$

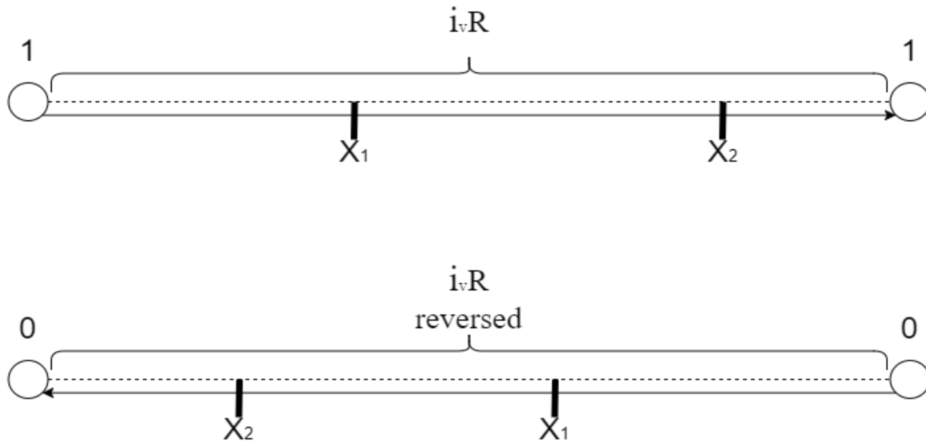
$$0 + 0 + 1 + 1 = (1,1) = 1$$

$$1 = 1$$

X. Ters Veta Numaraları

VA, V-1 - Ters Sayılar.

1	0
0	1
(0,1)	(1,0)
(1,0)	(0,1)
(0,0)	(1,1)
(1,1)	(0,0)



XI. Veta İşlev

İşlevler, varsayılan ve kaotik olmak üzere iki türe ayrılır. İlk işlev türü her zaman tek bir sayı "0" üretir, aksi takdirde "sakin işlev" olarak adlandırılabilir.

Varsayılan İşlev: f(x)vd = 1 * x + 0 ≈ 0

f((0,1))vd = 1*(0,1) + 0 = (1,1) + 0 = (1,0) ≈ 0

f(1)vd = 1*1 + 0 = 1 + 0 = (1,0) ≈ 0

İkinci tür işlevler, işlevde iki argümanın kullanılmasını içerir, bu, yeni bir veta numarası oluşturmanıza olanak tanır.

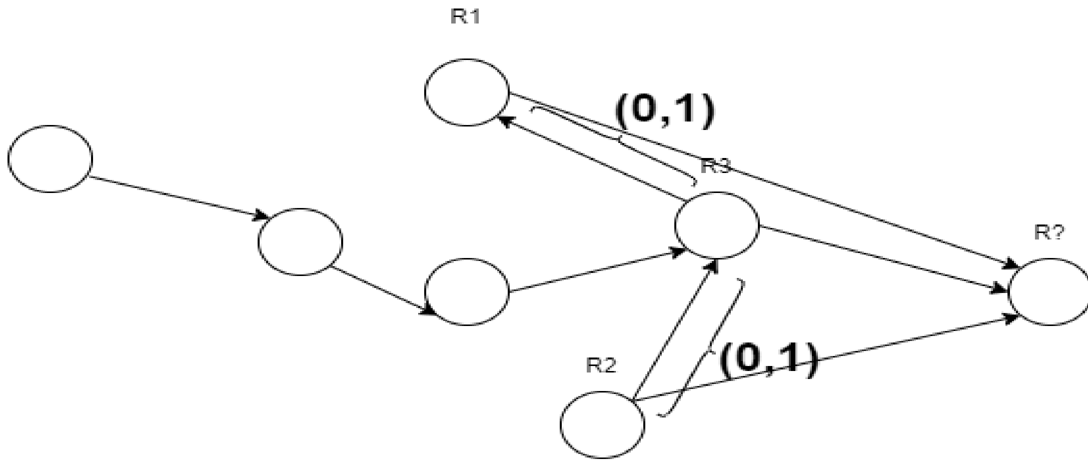
İki bağımsız değişkenli işlev, değiştirilebilen işlevdir: f(x)v

Örnek: f(x)v = (0,1) * x + (0,0)
f(x)v = (1,1) * x + (1,1)
f(x)v = (1,0) * x + (0,1)

XII. Veta Mantık

Mantık veta - benzer beyin süreçlerinin mantıksal süreçlerini ve olası diğer eylemlerin tam bir analizini tanımlayan bir mantık türüdür.

Örnek: R2 (0,1) ile R3'e gider,R3 (0,1) + (0,1) = (1,1) = 1 ile R1'e gider Bu, R'nin R1'e eşit olacağı anlamına gelir.

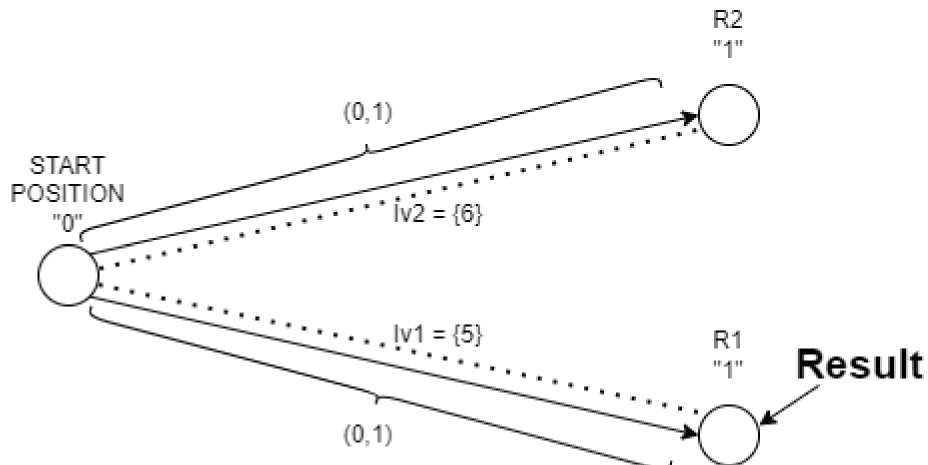


Örnek: Kitapları bir raftan diğerine kaydırmamızın teklif edildiğini, ilk rafta beş, ikinci rafta altı kitap olduğunu ve bize R1 R2 olayları için iki olası seçenek verildiğini hayal edin, bunların kayıtları aşağıdaki gibi olacaktır:

* Burada, kitap sayısının sürükleme başına mesafeye bölünmesiyle geçen sürükleme süresi olarak aralığı alıyoruz.

R1 = (0,1), Iv1 = {5}
R2 = (0,1), Iv2 = {6}

Aralık nedeniyle işlem daha az zaman alabileceğinden sonuç ilk olacaktır.



ÇÖZÜM

Bu teori, mantıksal verilerin hızlı dönüşümü ile hızlı bilgi işlemin yanı sıra yapay zekanın gelecekteki olanaklarını genişletecektir. Şu anda AI, geleneksel hesaplamaları kullanarak ve bulanık mantık sayesinde aktif olarak gelişiyor. Yapay zeka ve makine öğrenimi artık büyük bir keşfin eşiğinde, ancak her yerde verileri bağımsız olarak analiz etmeden ve bunları analiz ve tahmin için ortak bir aralığa sokmadan hazırlanmış kontroller var. Bu sebeplerden dolayı bu teori yapay zekanın karşılaşabileceği mantıksal ve istatistiksel problemlerin daha hızlı çözülmesine olanak sağlayacaktır.

KAYNAKÇA

1. David N. Osser, 2020, Psychopharmacology Algorithms
2. Thomas H. Cormen, Charles E. Leiserson, Ronald Rivest, Clifford Stein, 1989, Introduction to Algorithms
3. L. A. Zadeh, 1965, Information and Control 8 338-353, Fuzzy Sets
4. Daniel Mcneill, Paul Freiberger, 1994, Fuzzy Logic: The Revolutionary Computer Technology That Is Changing Our World
5. Luka Eciolaza, E. Trillias, 2015, Fuzzy Logic: An Introductory Course for Engineering Students
6. Michael Gr. Voskoglou, 2020, Fuzzy Sets, Fuzzy Logic and Their Applications
7. John Bird, 2010, Higher Engineering Mathematics
8. Yonina C. Eldar, 2022, Machine Learning and Wireless Communications
9. Pratap Dangeti, 2017, Statistics for Machine Learning
10. Maria Schuld, Francesco Petruccione, 2021, Machine Learning with Quantum Computers